

# Using the biofiles package

**Gerhard Schöfl**

Dept. of Bioinformatics  
DKMS Life Science Lab,  
Dresden, Germany

**November 21, 2017**

## Contents

1	The <i>biofiles</i> package . . . . .	1
1.1	Classes in the <i>biofiles</i> package . . . . .	1
1.2	Importing sequence and annotation data . . . . .	2
1.3	Saving and loading <i>gbRecords</i> . . . . .	4
2	Summarising a Genbank record . . . . .	4
3	Extracting subsets of the data . . . . .	5
3.1	Filtering the feature table . . . . .	6
3.2	Accessing annotation data . . . . .	8
4	Session Information . . . . .	12

## 1 The *biofiles* package

---

The *biofiles* package is an *R* package for interfacing with [GenBank](#) or [GenPept](#) flat file records. The GenBank sequence format is a rich data format for storing sequences and associated annotations. *biofiles* includes utilities for parsing and writing GenBank files, and useful methods for interacting with the annotation and sequence data contained in these files.

### 1.1 Classes in the *biofiles* package

The classes in *biofiles* attempt to encapsulate the information contained in GenBank files as faithfully as possible. The design is based heavily on the GenBank/Embl feature tables. The central class for storing feature annotation information is the S4 class *gbFeature*. Essentially, a *gbFeature* object contains a feature *key*, a short description of the type of feature (e.g. “CDS” or “gene”); the *location* of the feature on the sequence string, and *qualifiers*, additional information about the feature as key-value-pairs (e.g.: *locus\_tag*, *gene*, *product*, or *db\_xref* are frequently used qualifiers). A collection of *gbFeatures* is organised in a *gbFeatureTable*.

## Using the biofiles package

A *gbFeatureTable* together with metadata and sequence information stored as *XStringSet* objects (*Biostrings*) make up a *gbRecord*. A collection of *gbRecords* can be placed into a *gbRecordList*.

### 1.2 Importing sequence and annotation data

*gbRecords* are generated by importing “gb” or “gbk” files. These files can be either downloaded and imported, or they can be directly retrieved and imported using the NCBI’s public ENTREZ database server using the functionality provided the *reutils* package.

As an example *biofiles* includes the annotation file for the mitochondrial genome of *Saccharomyces cerevisiae*:

```
mito.path <- system.file("extdata", "S_cerevisiae_mito.gb", package="biofiles")
```

To read a GenBank file into *R* use the command

```
mito <- biofiles::gbRecord(mito.path)
mito
```

```
## Loading required package: Biostrings
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##   clusterExport, clusterMap, parApply, parCapply, parLapply,
##   parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##   Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##   as.data.frame, cbind, colMeans, colSums, colnames, do.call,
##   duplicated, eval, evalq, get, grep, grepl, intersect, is.unsorted,
##   lapply, lengths, mapply, match, mget, order, paste, pmax, pmax.int,
##   pmin, pmin.int, rank, rbind, rowMeans, rowSums, rownames, sapply,
##   setdiff, sort, table, tapply, union, unique, unsplit, which,
##   which.max, which.min
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
```

## Using the biofiles package

```
## The following object is masked from 'package:base':  
##  
##   expand.grid  
  
## Loading required package: IRanges  
## Loading required package: XVector  
  
##  
## Attaching package: 'Biostrings'  
  
## The following object is masked from 'package:base':  
##  
##   strsplit  
  
## An object of class 'gbRecord', with 102 features  
## LOCUS      NC_001224      85779 bp   DNA   circular PLN 05-JUN-2017  
## DEFINITION Saccharomyces cerevisiae S288c mitochondrion, complete genome.  
## ACCESSION  NC_001224  
## VERSION   NC_001224.1 GI:NA  
## DBLINK    Project: PRJNA128  
## KEYWORDS  RefSeq.  
## SOURCE    mitochondrion Saccharomyces cerevisiae S288C  
## ORGANISM  Saccharomyces cerevisiae S288C  
##           Eukaryota; Fungi; Dikarya; Ascomycota; Saccharomycotina;  
##           Saccharomycetes; Saccharomycetales; Saccharomycetaceae;  
##           Saccharomyces.  
## REFERENCE 1 (bases 1 to 85779)  
## AUTHORS   Foury,F., Roganti,T., Lecrenier,N. and Purnelle,B.  
## TITLE     The complete sequence of the mitochondrial genome of  
##           Saccharomyces cerevisiae  
## JOURNAL   FEBS Lett. 440 (3), 325-331 (1998)  
## PUBMED    9872396  
## REFERENCE 2 (bases 1 to 85779)  
## CONSRTM   NCBI Genome Project  
## TITLE     Direct Submission  
## JOURNAL   Submitted (14-JAN-2015) National Center for Biotechnology  
##           Information, NIH, Bethesda, MD 20894, USA  
## REFERENCE 3 (bases 1 to 85779)  
## CONSRTM   The Saccharomyces Genome Database  
## TITLE     Direct Submission  
## JOURNAL   Submitted (12-DEC-2014) Genetics, Saccharomyces Genome  
##           Database, Stanford University, Stanford, CA 94305, USA  
## REFERENCE 4 (bases 1 to 85779)  
## CONSRTM   MIPS  
## TITLE     Direct Submission  
## JOURNAL   Submitted (16-DEC-1998) Max-Planck-Institut fuer Biochemie, Am  
##           Klopferspitz 18a D-82152 Martinsried, Germany  
## COMMENT   PROVISIONAL REFSEQ: This record has not yet been subject to  
##           final NCBI review. The reference sequence is identical to  
##           KP263414. COMPLETENESS: full length.  
## ORIGIN    TTCATAATTAATTTTTTATATATATATTATATTATAATATTAATTTATATTATAAAAAATAATAT  
##           ...
```

## Using the biofiles package

```
##          TATATATATTGGAATAGTTATATTATTATACAGAAATATGCTTAATTATAATATAATATCCATA
## CONTIG
```

### 1.3 Saving and loading *gbRecords*

*gbRecord* objects can be stored efficiently by writing the content to a compressed “.rds” file using `saveRecord` and re-imported using `loadRecord`. Alternatively, *gbRecord* objects can be exported as a GenBank flat file using `write.GenBank` or in GenBank feature table format using `write.FeatureTable`.

```
biofiles::saveRecord(mito)
rm(mito)
mito <- biofiles::loadRecord("NC_001224.rds")
biofiles::summary(mito, n = 3)
```

```
## [[NC_001224]]
## 85779 bp: Saccharomyces cerevisiae S288c mitochondrion, complete genome.
## Id Feature Location GeneId Product Note
## 1 source 1..85779 NA NA NA
## 2 gene 731..802 NA NA NA
## 3 tRNA 731..802 NA tRNA-Pro NA
## ... .. ... .. ...
## 100 tRNA 85035..85112 NA tRNA-Met mitochondrial formylated methi ...
## 101 gene 85295..85777 RPM1 NA NA
## 102 ncRNA 85295..85777 RPM1 RPM1 RNA component of mitochondrial ...
```

## 2 Summarising a Genbank record

A quick overview over the content of a *gbRecord* is provided by `summary`:

```
biofiles::summary(mito)
## [[NC_001224]]
## 85779 bp: Saccharomyces cerevisiae S288c mitochondrion, complete genome.
## Id Feature Location GeneId Product ...
## 1 source 1..85779 NA NA ...
## 2 gene 731..802 NA NA ...
## 3 tRNA 731..802 NA tRNA-Pro ...
## 4 STS 2406..2759 NA NA ...
## 5 rep_origin complement(4012..4312) NA NA ...
## 6 gene 6546..8194 15S_RRNA NA ...
## 7 rRNA 6546..8194 15S_RRNA 15S ribosomal R ...
## ... .. ... .. ...
## 96 gene 79213..80022 c("COX3", "OXI2") NA ...
## 97 CDS 79213..80022 c("COX3", "OXI2") cytochrome c ox ...
## 98 rep_origin 82329..82600 NA NA ...
## 99 gene 85035..85112 NA NA ...
## 100 tRNA 85035..85112 NA tRNA-Met ...
```

## Using the biofiles package

```
## 101 gene      85295..85777      RPM1      NA      ...
## 102 ncRNA     85295..85777      RPM1     RPM1     ...
```

The frequency of features/qualifiers can be quickly tabulated:

```
biofiles::qualifTable(mito)

##      EC_number  codon_start      db_xref      gene  gene_synonym
##           9         19         194         42         14
##      locus_tag   mol_type  ncRNA_class      note  organelle
##           92          1          1         53          1
##      organism      product  protein_id  standard_name      strain
##           1          46         19          1          1
##      sub_strain  transl_table  translation
##           1          19          19

biofiles::featureTable(mito)

##
##      CDS      STS      gene      ncRNA      rRNA  rep_origin      source
##      19         1         46         1         2         8         1
##      tRNA
##      24
```

## 3 Extracting subsets of the data

There are a number of accessor methods for retrieving data from GenBank records.

We can access each component of the header, e.g.:

```
biofiles::getAccession(mito)
## [1] "NC_001224"

biofiles::getDefinition(mito)
## [1] "Saccharomyces cerevisiae S288c mitochondrion, complete genome."

biofiles::getGeneID(mito)
## [1] "NA"

biofiles::getOrganism(mito)
## [1] "Saccharomyces cerevisiae S288C"

biofiles::getLength(mito)
## [1] 85779

biofiles::getComment(mito)
## [1] "PROVISIONAL REFSEQ: This record has not yet been subject to final\nNCBI review. The reference sequence"
```

We can extract the DNA sequence as a whole as a *DNASTringSet*:

## Using the biofiles package

```
biofiles::getSequence(mito)

## A DNASTringSet instance of length 1
## width seq names
## [1] 85779 TTCATAATTAATTTTTTATATAT...TTAATTATAATATAATATCCATA NC_001224
```

We can extract the Feature table:

```
biofiles::ft(mito)

## 'gbFeatureTable' with 102 features:
## Feature:      Location/Qualifiers:
## source        1..85779
##               /organism = "Saccharomyces cerevisiae S288C"
##               /organelle = "mitochondrion"
##               /mol_type = "genomic DNA"
##               /strain = "S288C"
##               /sub_strain = "FY1679"
##               /db_xref = "taxon:559292"
## ...
## Feature:      Location/Qualifiers:
## ncRNA         85295..85777
##               /ncRNA_class = "RNase_MRP-RNA"
##               /gene = "RPM1"
##               /locus_tag = "Q0285"
##               /product = "RPM1"
##               /note = "RNA component of mitochondrial RNase P;
##               mitochondrial RNase P also contains the protein subunit
##               Rpm2p; RNase P removes 5' extensions from mitochondrial
##               tRNA precursors; RPM1 is conserved in bacteria, fungi, and
##               protozoa"
##               /db_xref = "GeneID:9164989"
##               /db_xref = "SGD:S000029023"
## Seqinfo:
## NC_001224 85779 DNA Saccharomyces cerevisiae S288c mitochondrion, comp ...
```

### 3.1 Filtering the feature table

The bulk of the data we are likely interested in is contained in the feature table. Features of interest can be filtered from a *gbRecord* or a *gbFeatureTable* using the function `filter`. For instance, to extract all "CDS" write:

```
cds <- biofiles::filter(mito, key = "CDS")
biofiles::summary(cds[1:2])

## Id Feature Location ...
## 12 CDS join(13818..13986,16435..16470,18954..18991,20508..20984,219 ...
## 14 CDS join(13818..13986,16435..16470,18954..18991,20508..20984,219 ...
```

A handy shortcut for filtering by feature key is provided by using the `[` operator:

## Using the biofiles package

```
cds <- mito["CDS"]
biofiles::summary(cds[3:4])

## Id Feature Location Ge ...
## 16 CDS join(13818..13986,16435..16470,18954..18991,20508..21935) AI ...
## 18 CDS join(13818..13986,16435..16470,18954..19996) AI ...
```

We can also filter features based on range:

```
f10000 <- biofiles::filter(mito, range = "..10000")
biofiles::summary(f10000)

## [[NC_001224]]
## 85779 bp: Saccharomyces cerevisiae S288c mitochondrion, complete genome.
## Id Feature Location GeneId Product Note
## 1 source 1..85779 NA NA NA
## 2 gene 731..802 NA NA NA
## 3 tRNA 731..802 NA tRNA-Pro NA
## 4 STS 2406..2759 NA NA NA
## 5 rep_origin complement(4012..4312) NA NA ORI1; m ...
## 6 gene 6546..8194 15S_RRNA NA NA
## 7 rRNA 6546..8194 15S_RRNA 15S ribosomal RNA ribosom ...
## 8 gene 9374..9447 NA NA NA
## 9 tRNA 9374..9447 NA tRNA-Trp mitocho ...
```

Or, perhaps more usefully, based on arbitrary qualifiers matching a regular expression. To find all CDSs for containing *cytochrome b* as a product, we use:

```
cytb <- biofiles::filter(mito, key = "CDS", product = "^cytochrome b$")
cytb

## An object of class 'gbRecord', with 1 features
## LOCUS NC_001224 85779 bp DNA circular PLN 05-JUN-2017
## DEFINITION Saccharomyces cerevisiae S288c mitochondrion, complete genome.
## ACCESSION NC_001224
## VERSION NC_001224.1 GI:NA
## DBLINK Project: PRJNA128
## KEYWORDS RefSeq.
## SOURCE mitochondrion Saccharomyces cerevisiae S288C
## ORGANISM Saccharomyces cerevisiae S288C
## Eukaryota; Fungi; Dikarya; Ascomycota; Saccharomycotina;
## Saccharomycetes; Saccharomycetales; Saccharomycetaceae;
## Saccharomyces.
## REFERENCE 1 (bases 1 to 85779)
## AUTHORS Foury,F., Roganti,T., Lecrenier,N. and Purnelle,B.
## TITLE The complete sequence of the mitochondrial genome of
## Saccharomyces cerevisiae
## JOURNAL FEBS Lett. 440 (3), 325-331 (1998)
## PUBMED 9872396
## REFERENCE 2 (bases 1 to 85779)
## CONSRTM NCBI Genome Project
## TITLE Direct Submission
## JOURNAL Submitted (14-JAN-2015) National Center for Biotechnology
```

## Using the biofiles package

```
## Information, NIH, Bethesda, MD 20894, USA
## REFERENCE 3 (bases 1 to 85779)
## CONSRTM The Saccharomyces Genome Database
## TITLE Direct Submission
## JOURNAL Submitted (12-DEC-2014) Genetics, Saccharomyces Genome
## Database, Stanford University, Stanford, CA 94305, USA
## REFERENCE 4 (bases 1 to 85779)
## CONSRTM MIPS
## TITLE Direct Submission
## JOURNAL Submitted (16-DEC-1998) Max-Planck-Institut fuer Biochemie, Am
## Klopferspitz 18a D-82152 Martinsried, Germany
## COMMENT PROVISIONAL REFSEQ: This record has not yet been subject to
## final NCBI review. The reference sequence is identical to
## KP263414. COMPLETENESS: full length.
## ORIGIN TTCATAAATTAATTTTTTATATATATATTATATTATAATATTAATTTATATTATAAAAAATAATAT
## ...
## TATATATATTGGAATAGTTATATTATTATACAGAAATATGCTTAATTATAATATAATATCCATA
## CONTIG
```

## 3.2 Accessing annotation data

After having filtered the features of interest we might want to access the various annotation data in a form more emenable to further analysis. Accessor functions include `start`, `end`, `span`, `strand`, `key`, `locusTag`, `geneID`, `product`, `dbxref`, `proteinID`, or `translation`:

```
biofiles::start(cds[1:3])

## [[1]]
## [1] 13818 16435 18954 20508 21995 23612 25318 26229
##
## [[2]]
## [1] 13818 16435 18954 20508 21995
##
## [[3]]
## [1] 13818 16435 18954 20508

biofiles::end(cds[1:3])

## [[1]]
## [1] 13986 16470 18991 20984 22246 23746 25342 26701
##
## [[2]]
## [1] 13986 16470 18991 20984 23167
##
## [[3]]
## [1] 13986 16470 18991 21935

biofiles::span(cds[1:3])

## [[1]]
## [1] 169 36 38 477 252 135 25 473
##
```



## Using the biofiles package

```
## [[2]]
## [1] 169 36 38 477 1173
##
## [[3]]
## [1] 169 36 38 1428
biofiles::strand(cds[1:3])
## [[1]]
## [1] 1 1 1 1 1 1 1 1
##
## [[2]]
## [1] 1 1 1 1 1
##
## [[3]]
## [1] 1 1 1 1
biofiles::locusTag(cds[1:3])
## [1] "Q0045" "Q0070" "Q0065"
biofiles::dbxref(cds[1:3])
## db_xref.GeneID db_xref.SGD
## 1 854598 S000007260
## 2 854597 S000007265
## 3 854596 S000007264
biofiles::product(cds[1:3])
## [1] "cytochrome c oxidase subunit 1"
## [2] "intron-encoded DNA endonuclease aI5 alpha"
## [3] "intron-encoded DNA endonuclease aI4"
biofiles::translation(cds[1:3])
## A AAStringSet instance of length 3
## width seq
## [1] 534 MVQRWLYSTNAKDIAVLYFMLAIFSGMAGTAMSL...TIFNLNTVKSSSIEFLLTSPPAVHSFNTPAVQS
## [2] 630 MVQRWLYSTNAKDIAVLYFMLAIFSGMAGTAMSL...VKRLFPMIYKYILPSMRYKFDIMLWQKKNMIN
## [3] 556 MVQRWLYSTNAKDIAVLYFMLAIFSGMAGTAMSL...EFYNLKELKAYNKSSDSMQYKAWLNFENKWKNK
```

We can easily extract the DNA sequences for a bunch of CDSs by simply passing a feature table to the function `getSequence`:

```
biofiles::getSequence(cds[1:6])
## A DNASTringSet instance of length 6
## width seq names
## [1] 1605 ATGGTACAAAGATGATTATATTC...ATACACCAGCTGTACAATCTTAA lcl|CDS.12|gb|NC...
## [2] 1893 ATGGTACAAAGATGATTATATTC...AAAAATATAATATGATTAATTAA lcl|CDS.14|gb|NC...
## [3] 1671 ATGGTACAAAGATGATTATATTC...AAAATAAATGAAAAATAAATA lcl|CDS.16|gb|NC...
## [4] 1248 ATGGTACAAAGATGATTATATTC...ATACATATTTAGAAAATAAATA lcl|CDS.18|gb|NC...
## [5] 2565 ATGGTACAAAGATGATTATATTC...AATATAATGGTCCAGGTTTATAA lcl|CDS.20|gb|NC...
## [6] 2505 ATGGTACAAAGATGATTATATTC...AAAATATAGGACCTGGTATATAA lcl|CDS.22|gb|NC...
```

## Using the biofiles package

A general function for accessing GenBank qualifiers is `qualif`. If no further argument is provided `qualif` returns a *data.frame* containing all qualifiers in the feature table. The argument `which` can be used to targeted specific qualifiers

```
biofiles::qualif(cds[1:3])

##      gene locus_tag gene_synonym EC_number
## 1    COX1    Q0045         OXI3    1.9.3.1
## 2  AI5_ALPHA    Q0070         <NA>    3.1.-.-
## 3     AI4     Q0065         <NA>    3.1.-.-
##
## 1          subunit I of cytochrome c oxidase (Complex IV); Complex IV is the terminal member of the mitoch
## 2                                     endonuclease I-SceIV; inv
## 3 endonuclease I-SceII; encoded by a mobile group I intron within the mitochondrial COX1 gene; intron is r
##  codon_start transl_table                                product
## 1           1           3          cytochrome c oxidase subunit 1
## 2           1           3 intron-encoded DNA endonuclease aI5 alpha
## 3           1           3          intron-encoded DNA endonuclease aI4
##  protein_id
## 1 NP_009305.1
## 2 NP_009306.1
## 3 NP_009307.2
##
## 1
## 2 MVQRWLYSTNAKDIAVLYFMLAIFSGMAGTAMSLIRLELAAPGSQYLHGNSQLFNVLVVGHAVLMIFFLVMPALIGGFGNYLLPLMIGATDTAFPRINNIAFV
## 3                                     MVQRWLYSTNAKDIAVLYFMLAIFSGMAGT
##  db_xref.GeneID db_xref.SGD
## 1           854598 S000007260
## 2           854597 S000007265
## 3           854596 S000007264

biofiles::qualif(cds[1:3], which = c("gene", "locus_tag", "EC_number", "product", "db_xref.GeneID"))

##      gene gene_synonym locus_tag EC_number
## 1    COX1         OXI3    Q0045    1.9.3.1
## 2  AI5_ALPHA         <NA>    Q0070    3.1.-.-
## 3     AI4         <NA>    Q0065    3.1.-.-
##
##                                     product db_xref.GeneID
## 1          cytochrome c oxidase subunit 1          854598
## 2 intron-encoded DNA endonuclease aI5 alpha          854597
## 3          intron-encoded DNA endonuclease aI4          854596
```

To transform all annotation (“key”, “location”, and “qualifiers”) into a *data.frame* we can use the function `select`:

```
cols <- c("key", "gene", "locus_tag", "product")
biofiles::select(cds[1:4], .cols = cols)

##  key      gene locus_tag                                product
## 1 CDS     COX1    Q0045          cytochrome c oxidase subunit 1
## 2 CDS  AI5_ALPHA    Q0070 intron-encoded DNA endonuclease aI5 alpha
## 3 CDS     AI4     Q0065          intron-encoded DNA endonuclease aI4
## 4 CDS     AI3     Q0060          intron-encoded DNA endonuclease aI3
```

## Using the biofiles package

Another useful format for annotation data are *GRanges* provided by the `bioconductor` package `GenomicRanges`. The function `ranges` allows to easily convert a `gbFeatureTable` into a *GRanges* object.

```
biofiles::ranges(cds)

## GRanges object with 50 ranges and 1 metadata column:
##           seqnames           ranges strand |           key
##           <Rle>             <IRanges> <Rle> | <character>
##  Q0045 NC_001224 [13818, 13986]      + |           CDS
##  Q0045 NC_001224 [16435, 16470]      + |           CDS
##  Q0045 NC_001224 [18954, 18991]      + |           CDS
##  Q0045 NC_001224 [20508, 20984]      + |           CDS
##  Q0045 NC_001224 [21995, 22246]      + |           CDS
##  ...      ...              ...      ... .           ...
##  Q0250 NC_001224 [73758, 74513]      + |           CDS
##  Q0255 NC_001224 [74495, 75622]      + |           CDS
##  Q0255 NC_001224 [75663, 75872]      + |           CDS
##  Q0255 NC_001224 [75904, 75984]      + |           CDS
##  Q0275 NC_001224 [79213, 80022]      + |           CDS
##  -----
##  seqinfo: 1 sequence (1 circular) from Saccharomyces cerevisiae S288c mitochondrion, complete genome. gen
```

A default `ranges` will only include the feature key as metadata and will expand compound locations (such as typical eukaryotic CDS). Compound locations can be merged using the argument `join = TRUE`, and more qualifiers can be included using the argument `include`:

```
biofiles::ranges(cds, join = TRUE, include = c("gene", "product", "db_xref"))

## GRanges object with 19 ranges and 5 metadata columns:
##           seqnames           ranges strand |           key           gene
##           <Rle>             <IRanges> <Rle> | <character> <character>
##  Q0045 NC_001224 [13818, 26701]      + |           CDS           COX1
##  Q0070 NC_001224 [13818, 23167]      + |           CDS      AI5_ALPHA
##  Q0065 NC_001224 [13818, 21935]      + |           CDS           AI4
##  Q0060 NC_001224 [13818, 19996]      + |           CDS           AI3
##  Q0055 NC_001224 [13818, 18830]      + |           CDS           AI2
##  ...      ...              ...      ... .           ...           ...
##  Q0140 NC_001224 [48901, 50097]      + |           CDS           VAR1
##  Q0160 NC_001224 [61022, 61729]      + |           CDS           SCEI
##  Q0250 NC_001224 [73758, 74513]      + |           CDS           COX2
##  Q0255 NC_001224 [74495, 75984]      + |           CDS           <NA>
##  Q0275 NC_001224 [79213, 80022]      + |           CDS           COX3
##
##                                     product db_xref.GeneID db_xref.SGD
##                                     <character> <character> <character>
##  Q0045          cytochrome c oxidase subunit 1          854598 S000007260
##  Q0070 intron-encoded DNA endonuclease aI5 alpha          854597 S000007265
##  Q0065          intron-encoded DNA endonuclease aI4          854596 S000007264
##  Q0060          intron-encoded DNA endonuclease aI3          854595 S000007263
##  Q0055          intron-encoded reverse transcriptase aI2          854594 S000007262
##  ...      ...              ...      ... .           ...           ...
##  Q0140 mitochondrial 37S ribosomal protein VAR1          854586 S000007275
##  Q0160          intron-encoded endonuclease I-SceI          854590 S000007279
```

## Using the biofiles package

```
## Q0250      cytochrome c oxidase subunit 2      854622 S000007281
## Q0255      maturase-like protein              854623 S000007282
## Q0275      cytochrome c oxidase subunit 3      854627 S000007283
## -----
## seqinfo: 1 sequence (1 circular) from Saccharomyces cerevisiae S288c mitochondrion, complete genome. gen
```

## 4 Session Information

---

All of the output in this vignette was produced under the following conditions:

```
utils::sessionInfo()

## R version 3.4.2 (2017-09-28)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 17.10
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=de_DE.UTF-8       LC_COLLATE=C
## [5] LC_MONETARY=de_DE.UTF-8   LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=de_DE.UTF-8      LC_NAME=C
## [9] LC_ADDRESS=C              LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4      parallel  stats      graphics  grDevices  utils      datasets
## [8] methods    base
##
## other attached packages:
## [1] Biostrings_2.46.0  XVector_0.18.0     IRanges_2.12.0
## [4] S4Vectors_0.16.0  BiocGenerics_0.24.0
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.13      knitr_1.17         magrittr_1.5
## [4] GenomicRanges_1.30.0  zlibbioc_1.24.0   foreach_1.4.3
## [7] GenomeInfoDb_1.14.0  stringr_1.2.0     highr_0.6
## [10] tools_3.4.2        htmltools_0.3.6   iterators_1.0.8
## [13] yaml_2.1.14       rprojroot_1.2     digest_0.6.12
## [16] assertthat_0.2.0    GenomeInfoDbData_0.99.1 bitops_1.0-6
## [19] codetools_0.2-15   Rcurl_1.95-4.8    evaluate_0.10.1
## [22] rmarkdown_1.8      stringi_1.1.6     compiler_3.4.2
## [25] biofiles_1.0.0     reutils_0.2.3     backports_1.1.1
## [28] BiocStyle_2.6.0
```